

## Лабораторная работа №2

### Ветвления и циклы в ассемблере.

#### Команда сравнения

Команда **CMP (Compare)** используется для сравнения двух операндов путем их вычитания и установки флагов в регистре EFLAGS в зависимости от результата сравнения. Команда имеет следующий синтаксис:

*cmp <операнд 1>, <операнд 2>*

Регистр **EFLAGS (Extended Flags Register)** в архитектуре x86 используется для хранения различных флагов, которые отражают состояние процессора и результаты выполнения различных инструкций. Эти флаги играют ключевую роль в управлении потоком выполнения программы, особенно при использовании условных инструкций.

Основные флаги **EFLAGS**:

- **ZF (Zero Flag)** устанавливается, если ОПЕРАНД1 равен ОПЕРАНДУ2
- **SF (Sign Flag)** устанавливается, если результат вычитания (ОПЕРАНД1 – ОПЕРАНД2) отрицателен. Используется для знаковых чисел.
- **CF (Carry Flag)** устанавливается, если происходит заем при вычитании, т.е. ОПЕРАНД1 меньше ОПЕРАНД2. Используется для беззнаковых чисел.
- **OF (Overflow Flag)** устанавливается, если происходит переполнение при вычитании знаковых чисел (например положительное число вычитается из отрицательного числа и результат превышает максимальное отрицательное значение)
- **PF (Parity Flag)** устанавливается, если количество единичных битов в младшем байте результата четно (используется для проверки четности результата)

#### Команды перехода

Команды перехода (или команды ветвления) в ассемблере используются для изменения порядка выполнения инструкций в программе. Они позволяют программе переходить к другой части кода в зависимости от определенных условий. Общий формат команд:

*jxx <метка>*

Основные виды команд перехода:

- **jmp (jump)** переход к указанной метке без проверки условий
- **je (jump if equal)** переход, если флаг равенства установлен **ZF = 1**
- **jne (jump if not equal)** переход, если флаг равенства не установлен **ZF = 0**
- **jg (jump if greater)** переход, если знаковое значение больше **ZF = 0** и **SF = OF**
- **jge (jump if greater or equal)** переход, если знаковое значение больше или равно **SF = OF**
- **jl (jump if less)** переход, если знаковое значение меньше **SF ≠ OF**
- **jle (jump if less or equal)** переход, если знаковое значение меньше или равно **ZF = 1** или **SF ≠ OF**
- **ja (jump if above)** переход, если беззнаковое значение больше **CF = 0** и **ZF = 0**

- **jae (jump if above or equal)** переход, если беззнаковое значение больше или равно **CF = 0**
- **jb (jump if below)** переход, если беззнаковое значение меньше **CF = 1**
- **jbe (jump if below or equal)** переход, если беззнаковое значение меньше или равно **CF = 1** или **ZF = 1**

Обычно команды перехода работают в паре с командой сравнения, например:

```

cmp %ebx, %eax;           # Сравнение eax и ebx
j1 less_than;             # Переход, если eax < ebx

# Если eax >= ebx
imul %ebx;                 # eax = eax * ebx
jmp exit;                  # Переход к метке exit

less_than:

# Если eax < ebx
sub %eax, %ebx;            # ebx = ebx - eax
mov %ebx, %eax;            # eax = ebx

exit:

# Завершение программы

```

## Циклы в ассемблере

Циклы в ассемблере представляют собой последовательности инструкций, которые повторяются определенное количество раз или до тех пор, пока выполняется определенное условие. В отличие от высокоуровневых языков программирования, где циклы могут быть записаны с помощью ключевых слов (например, `for`, `while`, `do...while`), в ассемблере циклы реализуются с помощью инструкций перехода.

Например найдём сумму всех цифр 1..9 с помощью условных переходов:

```

mov %eax, 0; # Инициализация суммы
mov %ecx, 1; # Инициализация счетчика

loop_start:

add %ecx, %eax;           # Добавляем значение счетчика к сумме
add $1, %ecx;             # Увеличиваем счетчик на 1
cmp %ecx, 10;             # Сравниваем счетчик с 10
j1 loop_start;            # Если счетчик меньше 10, повторить цикл

#Завершение программы

```

Команда **LOOP** в ассемблере является специализированной инструкцией для реализации циклов. Она используется для уменьшения значения регистра счетчика (обычно регистра CX или ECX) и выполнения перехода на указанную метку, если значение счетчика не равно нулю. Это делает её удобной для создания циклов с фиксированным количеством итераций. Формат:

**LOOP** <метка>

Перепишем предыдущую программу находящую сумму цифр с использованием команды LOOP:

```
mov %eax, 0; # Инициализация суммы
mov %ecx, 9; # Инициализация счетчика

loop_start:

    add %ecx, %eax; # Добавляем значение счетчика к сумме
    loop loop_start; # Уменьшаем счетчик и повторяем цикл

# если счетчик равен 0 выходим из цикла
# ...

#Завершение программы
```

### Контрольные вопросы:

- 1) Что такое команда сравнения в ассемблере? Для чего нужен регистр EFLAGS?
- 2) Какие флаги из регистра EFLAGS относятся к команде сравнения?
- 3) Что такое команды перехода, каков их формат, какие разновидности бывают?
- 4) Какими способами можно организовать цикл в ассемблере?
- 5) Формат объявления констант и переменных в языке C?
- 6) Что такое область видимости в языке C? Какие области видимости бывают?
- 7) Какие есть целочисленные типы данных в языке C?
- 8) Какие есть вещественные типы данных в языке C?
- 9) Что такое автоматическое приведение типов в языке C? Каковы правила автоматического приведения типов?
- 10) Что такое явное преобразование типа в языке C и как оно осуществляется?

### Задание

В соответствии с вариантом, необходимо рассчитать и вывести на экран значение, рассчитанное с помощью ассемблерной вставки применяя команды сравнения и перехода. Все переменные являются целочисленными и вводятся пользователем с клавиатуры.

#### Варианты:

1.  $Z = (A - B) * C$   
если  $Z < 0$ , то вычислить  $(A + C) * B - A$   
иначе вычислить  $C!$

2.  $Z = A + B * C$   
если  $Z \geq 0$ , то вычислить сумму квадратов чисел от A до B  
иначе вычислить  $C / (A - B) + A$
3.  $Z = A / B - C$   
если  $Z = 0$ , то вычислить  $(A + B) / (A - C)$   
иначе вычислить количество чисел от A до B кратных 3
4.  $Z = A / B * C$   
если  $Z \leq 0$ , то вычислить сумму всех положительных чисел от A до B  
иначе вычислить  $A * B + B / C$
5.  $Z = A * B - C$   
если  $Z < 0$ , то вычислить  $(B - C) * A + B$   
иначе выяснить является ли Z простым
6.  $Z = A + B - C$   
если  $Z \geq 0$ , то вычислить  $B / C + (A - B)$   
иначе посчитать сумму всех чисел от A до B кратных 7
7.  $Z = (A + B) / C$   
если  $Z \leq 0$ , то вычислить  $C / (A - B) - A$   
иначе найти B-е число Фибоначчи
8.  $Z = A * (B - C)$   
если  $Z = 0$ , то вычислить  $C * B / (A + B)$   
иначе вычислить  $A!$
9.  $Z = A / B - C$   
если  $Z < 0$ , то вычислить  $(A - B) * C + A$   
иначе вычислить произведение всех четных чисел от A до B
10.  $Z = A + B * C$   
если  $Z = 0$ , то вычислить произведение чисел от A до B кратных 3  
иначе вычислить  $(A - B) * C + A$
11.  $Z = (A - B) * C$   
если  $Z > 0$ , то вычислить количество чисел от A до B, кратных 4  
иначе вычислить  $C / (A + C) - B$
12.  $Z = A * B / C$   
если  $Z < 0$ , то вычислить сумму четных чисел от A до B  
иначе вычислить  $(C - B) * A + C$
13.  $Z = (A + B) / C$   
если  $Z = 0$ , то вычислить  $(A + B) - C / B$   
иначе вычислить количество чисел от A до B, кратных 6
14.  $Z = A * B * C$   
если  $Z \leq 0$ , то вычислить сумму всех чисел от A до B пропуская кратные 5  
иначе вычислить  $(A + B) * B / C$

15.  $Z = A - B + C$   
если  $Z > 0$ , то вычислить количество чисел от  $A$  до  $B$  кратных 7  
иначе вычислить  $B / C + A + C$

16.  $Z = (A + B) * C$   
если  $Z \geq 0$ , то вычислить  $B / C * (A - B)$   
иначе вычислить  $B!$

17.  $Z = A / (B + C)$   
если  $Z \leq 0$ , то вычислить  $A + C - B / C$   
иначе найти  $Z$ -е число Фибоначчи

18.  $Z = A * (B + C)$   
если  $Z = 0$ , то вычислить  $C / (A + B) + B$   
иначе вычислить  $(A - B)!$

19.  $Z = A / B + C$   
если  $Z < 0$ , то вычислить  $A - C / (A + B)$   
иначе проверить является ли  $Z$  простым числом

20.  $Z = A - B * C$   
если  $Z = 0$ , то вычислить сумму квадратов чисел от  $A$  до  $B$   
иначе вычислить  $(A - C) * B + A$

21.  $Z = (A - B) * C$   
если  $Z > 0$ , то вычислить количество чисел от  $A$  до  $B$ , кратных 5  
иначе вычислить  $C / (A - C) - B$

22.  $Z = A * B - C$   
если  $Z < 0$ , то вычислить  $B / C + A + C$   
иначе проверить является ли  $A$  простым числом

23.  $Z = (A + B) / C$   
если  $Z = 0$ , то вычислить  $(A + B) * C / B$   
иначе вычислить количество чисел от  $A$  до  $B$ , кратных 4

24.  $Z = A * B * C$   
если  $Z \leq 0$ , то вычислить сумму всех нечетных чисел от  $A$  до  $B$   
иначе вычислить  $A + B + B / C$

25.  $Z = A - B - C$   
если  $Z > 0$ , то вычислить  $Z!$   
иначе вычислить  $A + C - B / C$